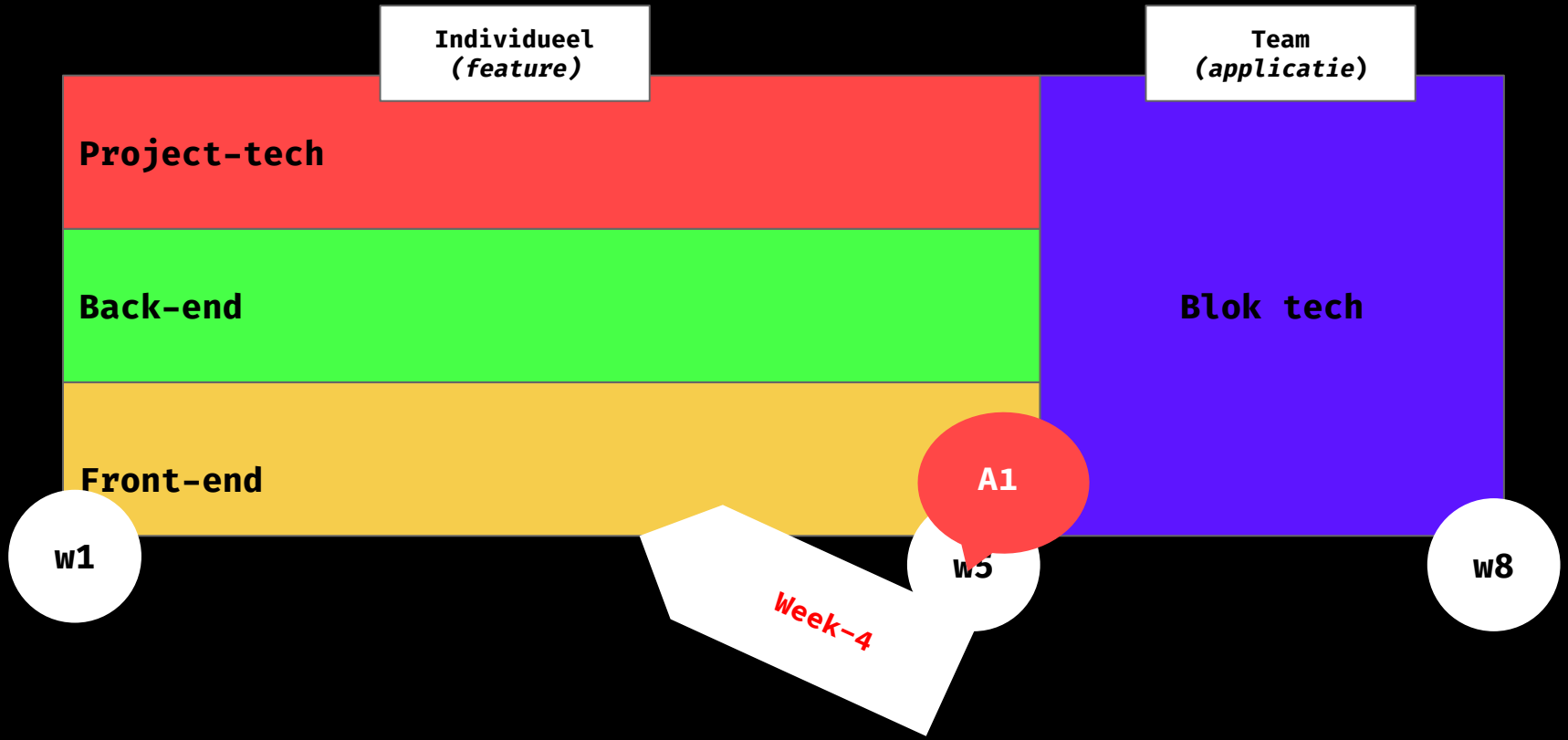
project-tech

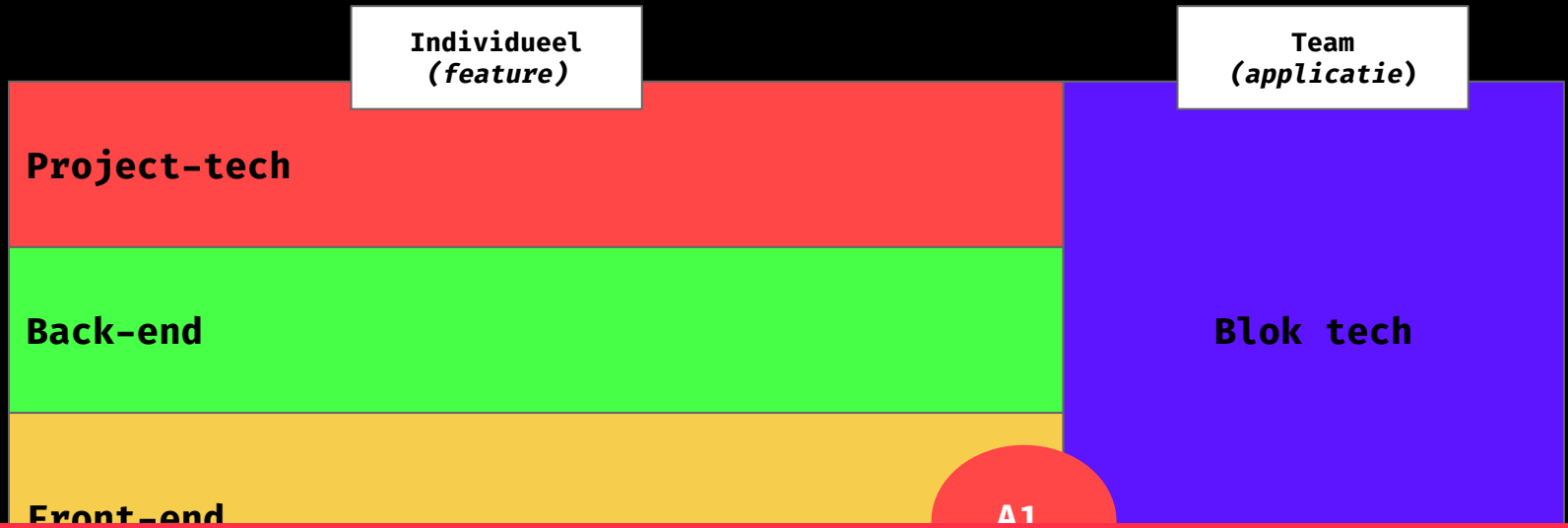
Refactor && Deployment

lab 4/8

Stand-up!

Show what you did

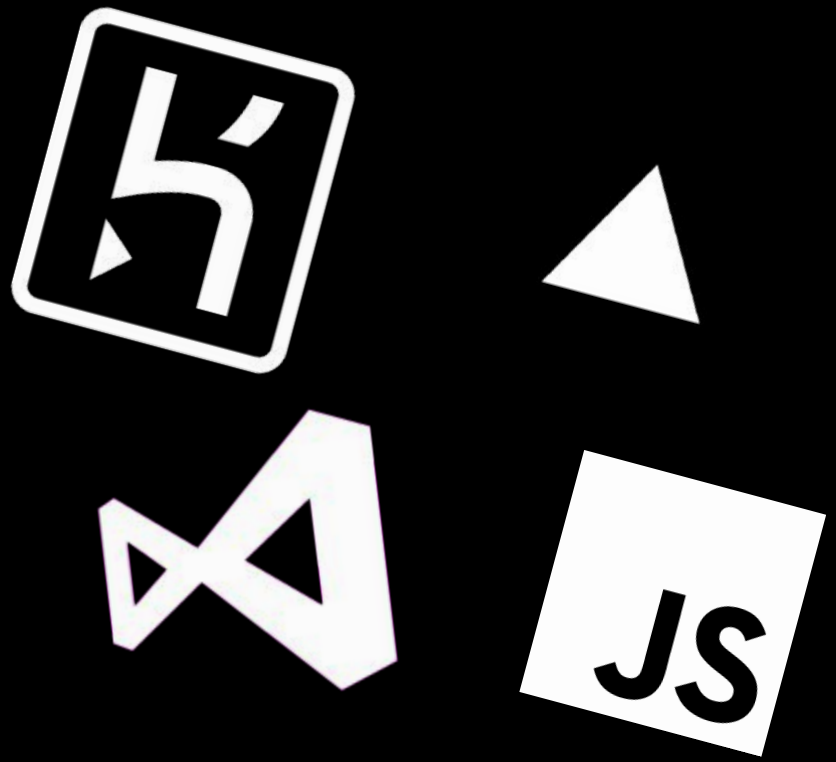

Individueel
(feature)
Team
(applicatie)
Project-tech
Back-end
Front-end
Blok tech
A1
w1
w5
w8
Week-4



Note: Next week (5) we'll do a final peer review.
See it as a checklist.

today

I. ~~Standup~~
II. Refactoring
III. Deployment



Refactoring

Refactor

?

[...] a disciplined technique for **restructuring an existing body of code**, altering its *internal structure* without changing its *external behavior*.

refactoring.com

Refactor

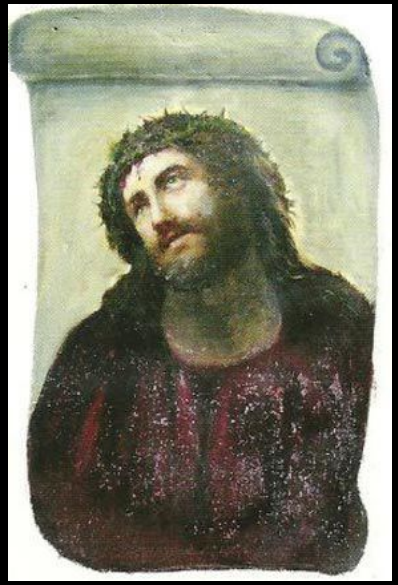
humans

The code you write will be *executed* by computers, but it will **exclusively be *read* by humans.** Therefore, it's critical that your code is easy to read, understand, and "mentally parse".

Refactor

humans

✨



‘It works!’

‘It’s beautiful!’

Refactor

example

Bad:

```
function addToDate(date, month) {
  // ...
}

const date = new Date();

// It's hard to tell from the function name what is added
addToDate(date, 1);
```

Good:

```
function addMonthToDate(month, date) {
  // ...
}

const date = new Date();
addMonthToDate(1, date);
```

Function names should say what they do

Refactor

example

Bad:

```
const DAYS_IN_WEEK = 7;
const daysInMonth = 30;

const songs = ["Back In Black", "Stairway to Heaven", "Hey Jude"];
const Artists = ["ACDC", "Led Zeppelin", "The Beatles"];
```

Good:

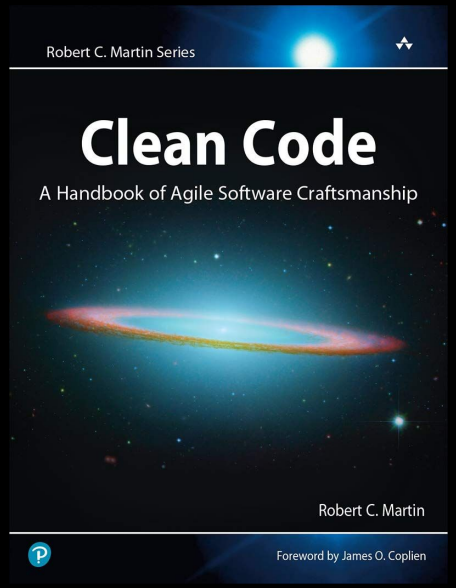
```
const DAYS_IN_WEEK = 7;
const DAYS_IN_MONTH = 30;

const SONGS = ["Back In Black", "Stairway to Heaven", "Hey Jude"];
const ARTISTS = ["ACDC", "Led Zeppelin", "The Beatles"];
```

Use consistent capitalization

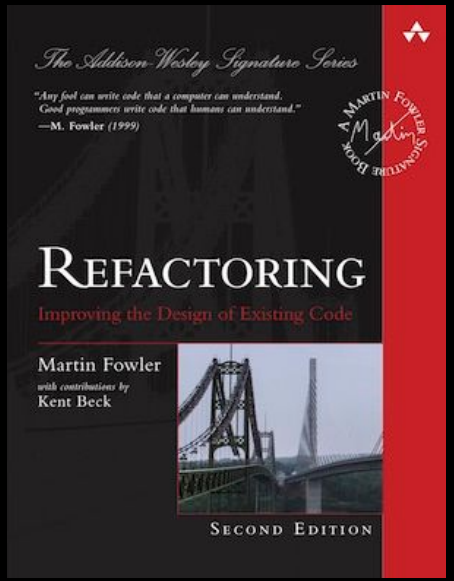
Refactor

books



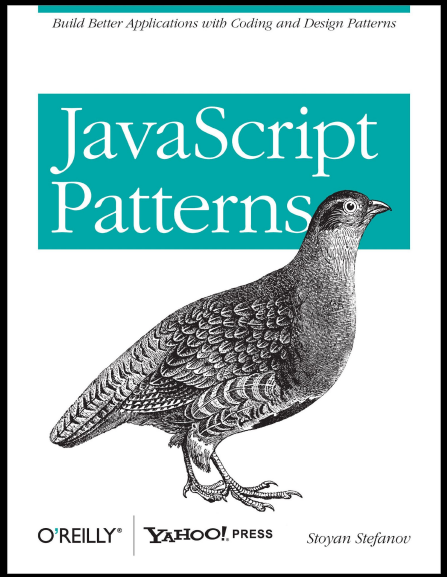
Clean Code

Robert C. Martin



Refactoring

Martin Fowler

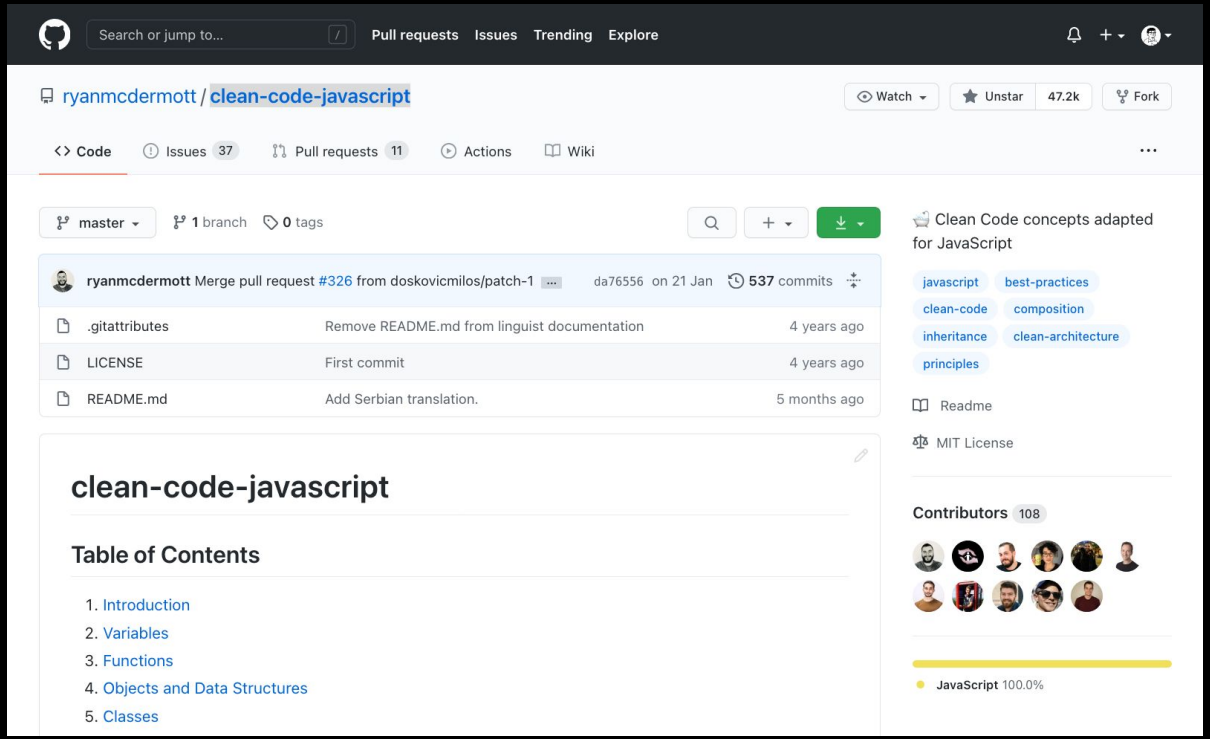


JS Patterns

Stoyan Stefanov

Refactor

github



github.com/clean-code-javascript

Refactor

standards

Code quality: how do we define “bad” code? If it’s overly complex? If it’s ‘messy?”

In conclusion: "bad" code can mean multiple things (simultaneously). Code can look horrible (to a developer), but still do what it is supposed to do. Is this bad code?

Refactor

standards

Why is maintaining a good coding style important?

- To avoid hard to catch errors as much as possible.
- Other developers can understand what your code does
- To save time and avoid stress
- Clean written code ***is*** your documentation!

Refactor

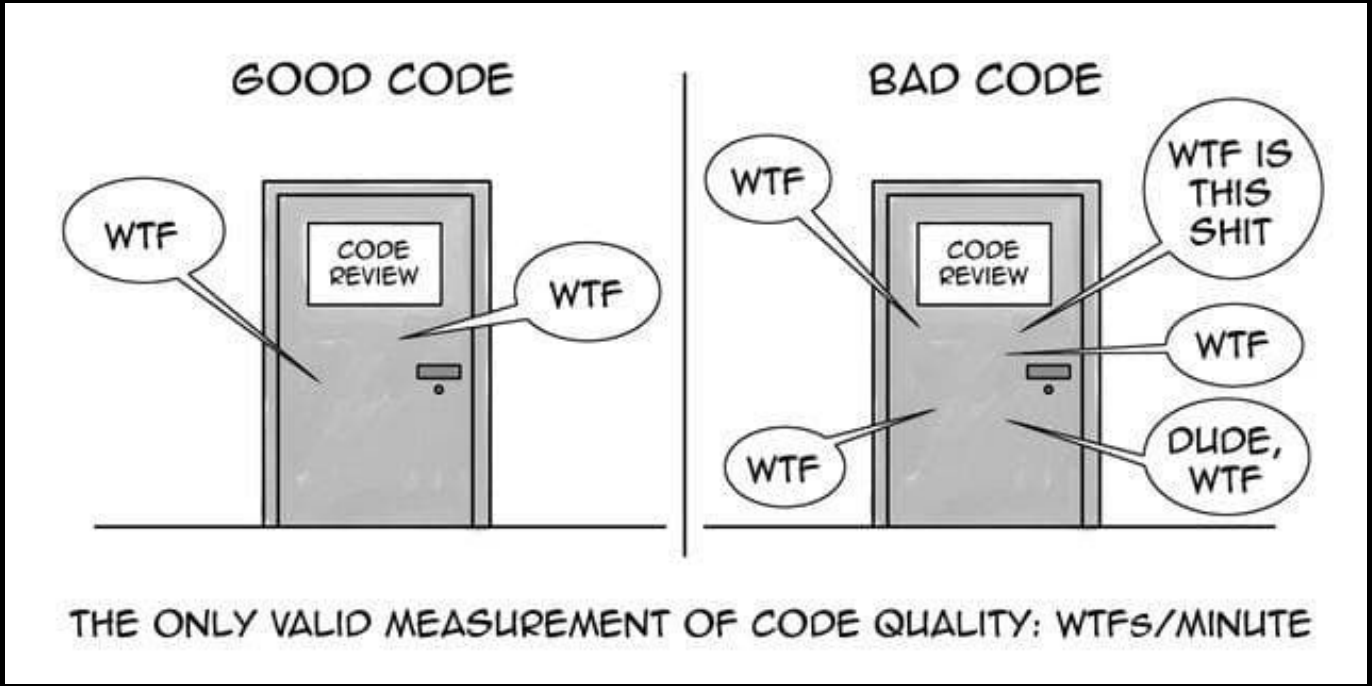
helpful tools

- Linters to enforce rules
- Formatters to format code
- A Good Night’s Sleep
- Code peer reviews

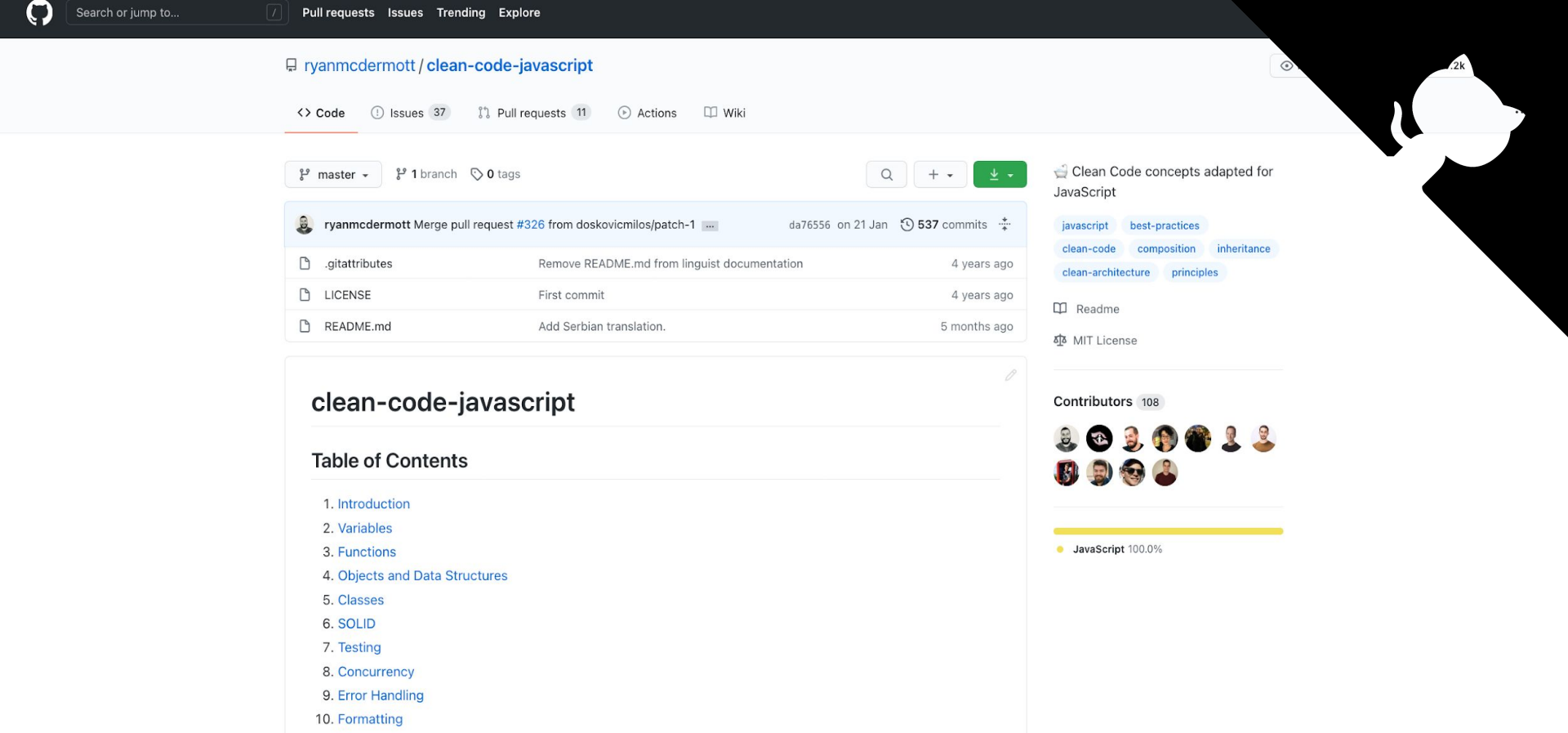
Refactor

low-hanging fruit

- ES5 vs. ES6 (*variables, arrow functions*)
- Old code *(that's in comments)*
- Inconsistent Indentation


GOOD CODE
WTF
CODE REVIEW
WTF
BAD CODE
WTF
WTF IS THIS SHIT
CODE REVIEW
WTF
WTF
DUDE, WTF
THE ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/MINUTE

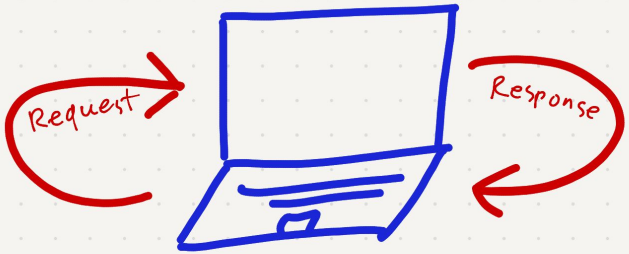
geekbot.com

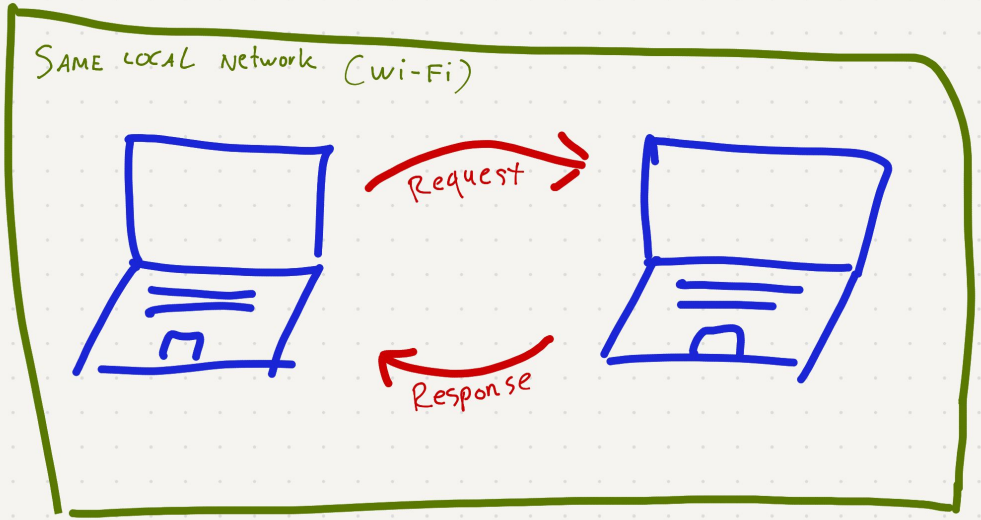


assignment **search for Refactors**

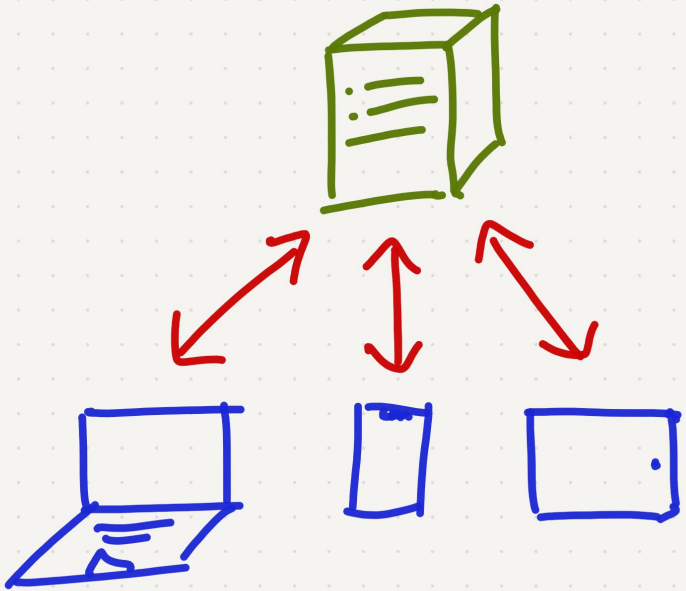
Deployment

Localhost:3000



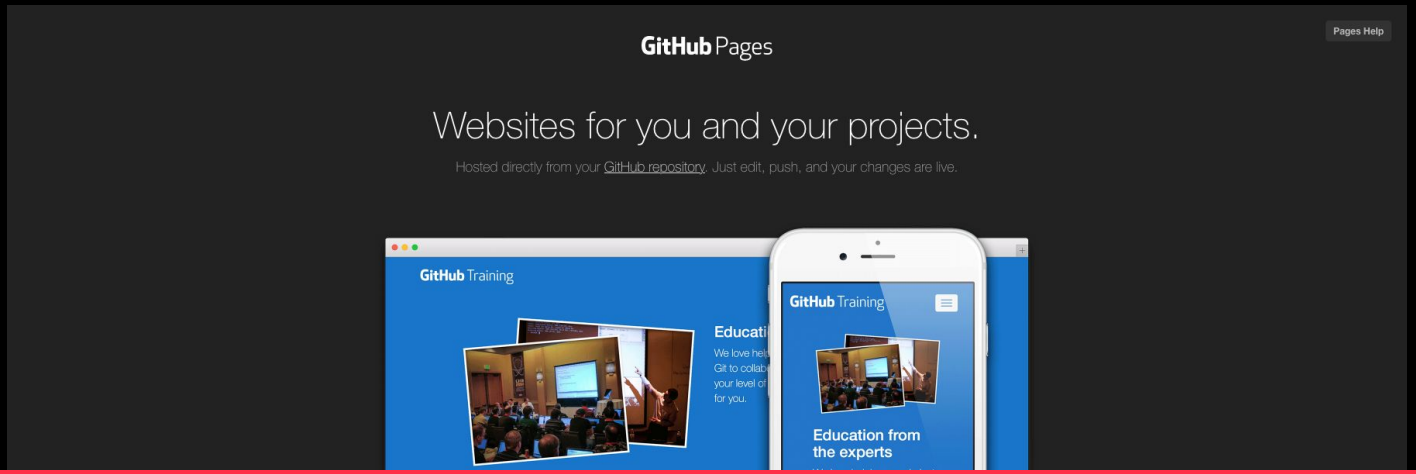


~~LOCALHOST~~ → 192.168.1.163



Deployment

static



Note: *Static hosting* like GitHub pages or Oege **won't work for dynamic Node.js** applications

Providers

There’s many hosting providers available, ranging from smaller (local) ones like Yourhosting to global conglomerates like Amazon Web-Services (AWS).

Choice is based on project requirements and costs.

Deployment

paas

🙂

Heroku

😰

DigitalOcean

VPS, PaaS, wtf

Many hosting providers offer virtual private servers (VPS). These “boxes” allow you to.. whatever you want. This is great if you have some very specific requirements, but also means you have to do *everything* yourself.

Popular options are TransIP (VPS), AWS (EC2), or DigitalOcean (Droplet).

VPS, PaaS, wtf

PaaS

There’s also dedicated services that do all the hard work for you. These are often a little more expensive and less-flexible.

We refer to these services as *Platform as a Service*.

Popular options are Vercel, Platform.sh, and DigitalOcean App Platform.

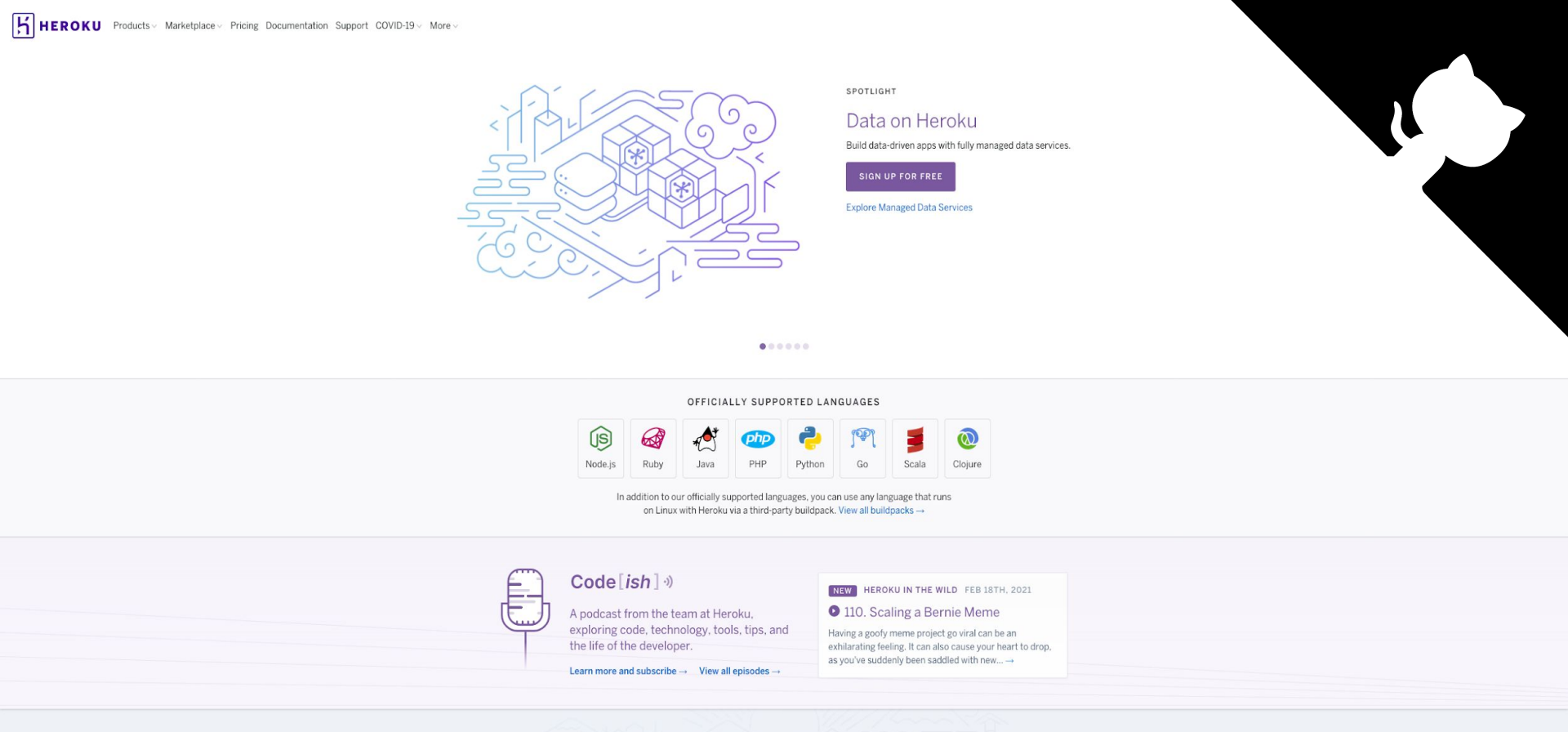
Deployment

```
// Files
mongodb-server/
├─ node_modules/
├─ static/
│  ├─ index.css
│  ├─ index.js
│  └─ upload/
├─ view/
│  ├─ add.ejs
│  ├─ detail.ejs
│  ├─ head.ejs
│  ├─ list.ejs
│  ├─ not-found.ejs
│  └─ tail.ejs
├─ .env
├─ index.js
└─ package.json
```

.env

.env

```
DB_HOST=localhost
DB_PORT=27017
DB_NAME=mymoviewebsite
```



Live demo **deploy, paas, hosting**

exit;

see you in lab-5!